REMARKS

This application has been reviewed in light of the Office Action dated April 17, 2007. Claims 1-12 and 22-30 are pending in the application. By the present amendment, claims 1, 2, 4, 5, 22, 23, 27, 28 and 30 have been amended. No new matter has been added. Claims 13-21 have been canceled without prejudice. The Examiner's reconsideration of the rejection in view of the amendment and the following remarks is respectfully requested.

By the Office Action, claims 1-30 stand rejected under 35 U.S.C. §102 (b) as being anticipated by U.S. Patent No. 5,765, 035 to Tran (hereinafter Tran).

Claims 1 and 22 have been amended to further clarify the invention.

Claim 1 (and essentially claim 22) now recites, *inter alia*, a pointer dependency checking stage located upstream of the pointer register stage, which determines if instruction pointer dependencies exist and stalls an instruction if necessary to resolve inter-instruction dependencies in the pointer register stage; a dependence checking stage located downstream of the pointer register file and configured to perform checking on instructions and stalling an instruction if necessary to resolve inter-instruction dependencies in an instruction register file; an issue stage located downstream from the dependence checking stage; the instruction register file configured to store instruction information and updates; at least one pointer functional unit providing pointer information updates directly to the pointer register stage and/or directly to an input of the at least one pointer functional unit as a pointer bypass such that the pointer information is processed and updated; and at least one instruction functional unit providing information updates directly to the instruction register stage and/or directly to an input of the at least one instruction functional unit as an instruction bypass such that instruction update information is processed and updated and wherein instructions updating the instruction register

file and instructions updating the pointer register file are simultaneously processed to track and keep current pointer and dependency updates between instructions.

Tran is directed to a system having a reorder buffer that detects dependencies between accesses to caches. Tran does not disclose or suggest the stages or operations as set forth in the present claims.

According to the present invention, one deficiency of the prior art implementations of an indirect register access mechanism is that no checking is performed for register dependencies between instructions (that is, dependencies through registers in a register file). These dependencies cannot be checked before the pointers are accessed, because the values of indexes to the register file are not known at the dependence checking stage, and different pointers may point to the same entry in the register file (i.e., aliasing problem).

The present claims make the pointer updates to the pointer register file available to the instruction in subsequent instructions with low latency (the latency could be as low as a single cycle), resulting in a reduced frequency of bubbles (or reduced number of unused issue slots) in instruction sequences with frequent inter-instruction dependencies through register pointers. By running through the pointer information in advance of instruction issuance, delays in the pipeline can be reduced.

Tran does not address this problem or provide the structure to address this problem. Tran does not disclose or suggest a pointer register stage which stores pointer information and updates; a pointer dependency checking stage located upstream of the pointer register stage, which determines if instruction pointer dependencies exist and stalls an instruction if necessary to resolve inter-instruction dependencies in the pointer register stage; and a dependence checking stage located downstream of the pointer register file and configured

to perform checking on instructions and stalling an instruction if necessary to resolve inter-instruction dependencies in an instruction register file.

The operational stages and their functions in accordance with the present claims are not disclosed or suggested by Tran. In accordance with claim 1, pointer dependence is checked followed by a pointer register file. Then instruction dependence is checked followed by an issue stage and then an instruction register file. This sequence and the benefits provided by the present principles are not disclosed or suggested by Tran.

In addition, Tran does not include other elements as set forth in the claims. For example, Tran fails to disclose or suggest AT LEAST:

at least one pointer functional unit providing pointer information updates directly to the pointer register stage and/or directly to an input of the at least one pointer functional unit as a pointer bypass such that the pointer information is processed and updated; and

at least one instruction functional unit providing information updates directly to the instruction register stage and/or directly to an input of the at least one instruction functional unit as an instruction bypass such that instruction update information is processed and updated and wherein instructions updating the instruction register file and instructions updating the pointer register file are simultaneously processed to track and keep current pointer and dependency updates between instructions.

Tran does not disclose or suggest a pointer functional unit. In addition, Tran does not provide such a pointer functional unit that provides pointer information updates directly to the pointer register stage and/or directly to an input of the at least one pointer functional unit as a pointer bypass such that the pointer information is processed and updated. Further, the functional units described by Tran do not disclose or suggest an instruction

functional unit providing information updates directly to the instruction register stage and/or directly to an input of the at least one instruction functional unit as an instruction bypass.

Tran does not provide that instruction update information is processed and updated wherein instructions updating the instruction register file and instructions updating the pointer register file are simultaneously processed to track and keep current pointer and dependency updates between instructions.

It is respectfully submitted that claims 1 and 22 are in condition for allowance over Tran for at least the stated reasons. Reconsideration of the rejection is earnestly solicited.

Dependent claims 2-12 and 23-30 are also believed to be allowable for at least their dependencies from claims 1 and 22, respectively. Reconsideration of the rejection is earnestly solicited. Other reasons exist for allowing the dependent claims as well. The Examiner is respectfully requested to review these claims again, and reconsider the rejections.

The Applicant notes with appreciation, the *Response to Arguments* section, and the time that the Examiner dedicated to providing an explanation of the rejection. However, in view of the foregoing amendments and remarks, it is respectfully submitted that all the claims now pending in the application are in condition for allowance. Early and favorable reconsideration of the case is respectfully requested.

It is believed that no additional fees or charges are currently due. However, in the event that any additional fees or charges are required at this time in connection with the application, they may be charged to applicant's IBM Deposit Account No. 50-0510.

Respectfully submitted,

Date: 7/10/07

By: 
James J. Bitetto
Registration No. 40,513

Mailing Address:

KEUSEY, TUTUNJIAN & BITETTO, P.C.
20 Crossways Park North, Suite 210
Woodbury, NY 11797
Tel: (516) 496-3868
Fax: (516) 496-3869